Title of the invention

Method for managing a set of alarms emitted by sensors for detecting intrusions of an information security system

Background of the invention

The invention relates to a method of managing alerts issued by intrusion detection sensors.

The security of information systems relies on the deployment of intrusion detection systems (IDS) including intrusion detection sensors that send alerts to alert management systems.

Intrusion detection sensors are active components of the intrusion detector system that analyze one or more sources of data for events characteristic of an intrusive activity and send alerts to an alert management system that centralizes the alerts from the various sensors and optionally analyzes all the alerts.

Intrusion detection sensors generate a very large number of alerts, possibly several thousand alerts a day, as a function of the configuration and the environment.

The surplus alerts may result from a combination of several phenomena. First of all, false alerts represent up to 90% of the total number of alerts. Secondly, it is often the case that alerts are too "granular", i.e. that their semantic content is highly impoverished. Finally, alerts are often repetitive and redundant.

The surplus alerts therefore make it difficult for a human security operator to understand them and manipulate them.

To facilitate analysis by a security operator, it is therefore necessary to process alerts upstream of the management system.

Existing alert management systems store the alerts in a relational database management system (RDBMS). The security operator can interrogate the RDBMS by submitting to it a request relating to the properties of the alerts.

The RDBMS responds by supplying to the operator all the alerts whose description matches the request.

The drawback of those systems is that many granular alerts may be supplied to the operator, which makes analyzing them a painstaking task.

Object and summary of the invention

An object of the invention is to eliminate the above drawbacks and to provide a simple method of managing alerts issued by intrusion detection sensors to enable flexible, easy, and fast consultation of the alerts.

The above objects are achieved by a method of managing alerts issued by intrusion detection sensors of an information security system including an alert management system, each alert being defined by an alert identifier and an alert content, which method is characterized in that it includes the following steps:

· associating with each of the alerts issued by the intrusion detection sensors a description including a conjunction of valued attributes belonging to attribute domains;

· organizing the valued attributes belonging to each attribute domain into a taxonomic structure defining generalization relationships between said valued attributes, the plurality of attribute domains thus forming a plurality of taxonomic structures;

· completing the description of each of said alerts with sets of values induced by the taxonomic structures on the basis of the valued attributes of said alerts to form complete alerts; and

· storing said complete alerts in a logic file system to enable them to be consulted.

Thus storing complete alerts in a logic file system enables a security operator to consult the alert management system efficiently, quickly, and flexibly, in order to obtain a precise view of all the alerts issued by the intrusion detection sensors.

Complete alerts may be consulted by successively interrogating and/or browsing said complete alerts so that the alert management system responds to a request by supplying pertinent valued attributes enabling a subset of complete alerts to be distinguished in a set of complete alerts satisfying the request in order to enable said request to be refined.

The pertinent valued attributes assigned the highest priority are preferably those that are most general, given the taxonomic structures.

The alert management system advantageously further responds to the request by supplying alert identifiers satisfying the request and whose description cannot be refined with respect to said request.

The alert identifier is a pair consisting of an identifier of the intrusion detection sensor that produces the alert and an alert serial number assigned by said sensor.

The content of each alert includes a text message supplied by the corresponding intrusion detection sensor.

Each valued attribute includes an attribute identifier and an attribute value.

According to one aspect of the invention, each attribute identifier is associated with one of the following attribute domains: attack domain, attacker identity domain, victim identity domain, and attack date domain.

The description of a given alert is advantageously completed by recovering recursively from generalization relationships of the taxonomic structures a set including the more general valued attributes not already included in the description of another alert completed previously.

According to one particular aspect of the invention, the valued attributes in the taxonomic structure are organized in accordance with an acyclic directed graph.

The invention also provides a computer program designed to execute the above method when it is executed by the alert management system.

Brief description of the drawings

Other features and advantages of the invention will emerge on reading the description given below by way of illustrative and non-limiting example and with reference to the appended drawings, in which:

- Figure 1 is a diagram of an information security system including an alert management system of the invention;
- Figure 2 is a flowchart showing the steps of the method of the invention of managing alerts;
- Figure 3A shows one example of documentation associated with attack signatures; and
- Figure 3B is a diagram of a taxonomic structure associated with the Figure 3A example.

Detailed description of embodiments of the invention

Figure 1 shows one example of an intrusion detection system 1 connected via a router 3 to an external network 5 and to a distributed architecture internal network 7a, 7b.

The intrusion detection system 1 includes intrusion detection sensors 11a, 11b, 11c and an alert management system 13. A first intrusion detection sensor 11a monitors alerts coming from outside, a second intrusion detection sensor 11b monitors a portion 7a of the internal network comprising workstations 15, and the third intrusion detection sensor 11c monitors another portion 7b of the internal network comprising servers 17 supplying information to the external network 5.

The alert management system 13 includes a host 19 dedicated to processing alerts, a database 21, and an output unit 23.

The logic file system may be of the LISFS type proposed by Padioleau and Ridoux in their paper "A Logic File System", Usenix Annual Technical Conference 2003.

In the LISFS logic file system, files are objects associated with descriptions expressed in the form of propositional logic. The description of a file is a conjunction of properties.

The properties of the files are the directories of the file system and the path of a file is therefore its description. A path is therefore a logic formula. A location in the file system contains all the files whose description satisfies the formula corresponding to the path of the location.

As in a conventional file system, specific commands are used to browse and manipulate files and their descriptions.

As indicated by the arrows 26, the sensors 11a, 11b, 11c of the intrusion detection system 1 send alerts 25 to the alert management system 13 which, in accordance with the invention, manages these alerts and stores them in the logic file system 21 to enable flexible consultation thereof via the output unit 23.

The host 19 of the alert management system 13 comprises processor means for managing alerts in this way.

The alert management system may execute a computer program designed to implement the present invention.

Figure 2 is a flowchart showing the steps of the method of the invention for managing a set O of alerts issued by intrusion detection sensors.

Each alert $\underline{o}$ of this set of alerts is defined by an alert identifier and an alert content.

An alert $o \in O$ can be defined by a single alert identifier id(o) given by a pair (s,n) in which $\underline{s}$ is the serial identifier of the intrusion detection sensor that produces the alert and $\underline{n}$ is the alert serial number assigned by that sensor to the alert $\underline{o}$.

The content m_o of the alert o includes a text message addressed to the security operator and supplied by the intrusion detection sensor that produced the alert.

A step E1 associates with each of the alerts issued by the intrusion detection sensors 11a, 11b, 11c a description d(o) including a conjunction of a plurality of valued attributes $\{d_{o,i}\}$ belonging to a plurality or a set of attribute domains {A}.

Thus a description d(o) of an alert is a conjunction of p valued attributes, i.e. $d(o) = d_{o,1} \wedge \cdots \wedge d_{o,p}$.

A valued attribute $d_{o,i}$ is a pair (a,ν) comprising an attribute identifier a and an attribute value ν.

Each attribute identifier a is associated with an attribute domain A from the following domains: attack domain, attacker identity domain, victim identity domain, and attack date domain.

Generally speaking, an attribute domain A is formed of a discrete set having a partial order relationship $\prec_A$ defining the domain of the valued attribute $d_{o,i}$.

A step E2 organizes the valued attributes $d_{o,i}$ belonging to each attribute domain A into a taxonomic structure defining generalization (or specialization) relationships between those valued attributes. There is a taxonomy for each attribute domain. Thus the plurality of attribute domains forms a plurality of taxonomic structures.

The generic taxonomic structure of the valued attributes is an acyclic directed graph.

The taxonomic relationships are modeled by axioms. Accordingly, a valued attribute d more specific than another valued attribute d′ is modeled by an axiom $d \models d'$, i.e. the valued attribute d′ is a logical consequence of the valued attribute d. In other words, an alert that has the specific valued attribute d automatically has the less specific valued attribute d′.

A step E3 completes the description of each of the alerts issued by the intrusion detection sensors 11a, 11b, 11c with sets of values induced by the taxonomic structures based on the valued attributes of those initial alerts to form complete alerts.

The valued attributes of the alerts produced by the intrusion detection sensors are the most specific of the taxonomies.

Accordingly, on receiving a given alert, the alert management system 13 can, for example, complete the description of that alert by recursively recovering from the generalization relationships of the taxonomic structures a set including the more general valued attributes that have not already been included in the description of another alert previously completed.

In other words, the description of a given alert is completed by a process that consists in working back through a given taxonomy starting from a given valued attribute. If a valued attribute exists already in the description of another alert processed previously, then this process is stopped; if not, it is added, and the process is iterated from this added valued attribute.

There follows an example of an algorithm "*CompleteDescription*" describing a process for completing the description of an alert.

```
        CompleteDescription
    if there is no $d_{o,i}$ do
        $D = \{d'_{o,i} : d_{o,i} \models d'_{o,i}\}$
        for each $d'_{o,i} \in D$ do
            CompleteDescription ($d'_{o,i}$)
         do
         mkdir $d'_{o,1} / \cdots / d'_{o,n}$
    do
```

This algorithm first tests the existence of a given valued attribute $d_{o,i}$. If that attribute $d_{o,i}$ does not exist, the set $D = \{d'_{o,i} : d_{o,i} \models d'_{o,i}\}$ of valued attributes that are more abstract, given the taxonomies, is recovered. Then, for each element $d'_{o,i}$ belonging to D, the

CompleteDescription algorithm is called recursively. At the end, the valued attribute is added to the alert management system by the command "*mkdir*" of the LISFS logic file system.

Finally, a step E4 in Figure 2 stores the alerts completed in the preceding step in the logic file system 21 to enable them to be consulted.

Below is an example of a "*StoreAlert*" algorithm describing a process for storing a new alert in an LISFS logic file system.

```
          StoreAlert
      For each $d_{o,i}$ do
          CompleteDescription ($d_{o,i}$)
      do
      cp $m_o\, d_{o,1}/\cdots/d_{o,n}/a$
```

For each description element $d_{o,i}$ the above algorithm completes a given alert o iteratively by calling the "*CompleteDescription*" algorithm described above.

When all the description elements of the given alert have been completed, the complete alert and its content are stored by a store command "cp" with parameters consisting of the alert content m_o, the alert description $d_{o,1}/\cdots/d_{o,n}$, and the alert identifier a.

Storing the complete alerts in the logic file system 21 enables them to be consulted and/or browsed successively in the set of complete alerts.

Accordingly, in response to a request from a security operator, and in order to refine the request, the alert management system 13 supplies pertinent valued attributes for distinguishing a subset of complete alerts in a set of complete alerts satisfying the request.

A request from the security operator is a logic formula f which combines conjugations (AND) $\wedge$, disjunctions (OR) $\vee$, and negations (NOT) $\neg$ of valued attributes.

Generally speaking, the description d(o) of an alert o satisfies a request f if the request f is a logical consequence of the description d(o). The set of alerts satisfying the request f, called the extension of f, is therefore $ext(f) = \{ o \in O \ : d(o) \models f \}$.

The set A of pertinent valued attributes is the set of valued attributes belonging to valued attribute domains A, such that for any pertinent valued attribute p of A, the set of complete alerts satisfying the conjunction of the current request f with the pertinent valued attribute p is strictly contained in the set of complete alerts satisfying the current request f. Accordingly, this set A of pertinent valued attributes enabling alerts to be distinguished from each other can be defined as follows:

$$A = \{ p \in A \ : \phi \subset ext(f \wedge p) \subset ext(f) \}.$$

The set A may be considered as a set of browsing links by defining each pertinent value attribute p as a browsing link. The security operator can therefore refine a current request f by choosing a browsing link $p \in A$ supplied by the alert management system 13. The current request f from the security operator is thus transformed into a new request $f \wedge p$.

To reduce further the number of responses, the alert management system 13 advantageously gives priority to supplying the pertinent valued attributes that are the most general in regard to the plurality of taxonomic structures.

The set A_{max} of the most general pertinent valued attributes is then given by the set $\max_{\models}(A)$, which may be defined as follows:

$$\max_{\models}(A) = \{ p \in A \ : \text{there exists no } p' \in A, p' \neq p, p \models p' \}.$$

Accordingly, this set $\max_{\models}$ (A) is the set of all pertinent valued attributes p of A that do not have a more general valued attribute.

Moreover, in response to the current request f, the alert management system supplies a set O of alert identifiers the description whereof satisfies the current request f and cannot be refined, i.e. described more precisely, with respect to that request $\underline{f}$. Accordingly, the set O of alert identifiers includes all alert identifiers whose description satisfies the current request $\underline{f}$ and such that there exists no pertinent valued attribute $\underline{p}$ such that the conjunction of $\underline{f}$ and $\underline{p}$ is satisfied by the description of that same alert. Accordingly, the set O can be defined as follows: $O = \{ id(o) \ : o \in \ O, \ d(o) \models f$, and there exists no $p \in A$ with $d(o) \models f \wedge p \ \}$.

Note that the logic file system 21 (e.g. the LISFS file system) provides commands for browsing objects (command "cd"), interrogating objects (command "ls") and storing objects (commands "cp" and "mkdir").

For example, in LISFS, a request that retrieves alerts whose victim is a web proxy and whose attacker is not internal is expressed as follows:

ls / "web proxy victim"/! "internal attacker".

Generally speaking, an alert coming from an intrusion detection sensor is a set of four valued attributes: *attack, attacker, victim*, and *date*.

The domain of the attack valued attribute consists of the identifiers of attack signatures in alerts generated by the intrusion detection sensors 11a, 11b, 11c.

The domain of the attack valued attribute also includes any vulnerabilities exploited by an attack. These vulnerabilities are more abstract, i.e. more general, than the attack identifiers.

The other values used to qualify the attacks are produced from keywords employed to qualify the attacks in the documentation of the intrusion detection sensors 11a, 11b, 11c.

Consider, by way of example, the Snort™ sensor and the "msg" field of the signature documentation.

Figure 3A shows an example of documentation associated with attack signatures.

Column 31 of table 33 includes integers designating attack signatures. Column 35 of table 33 includes documentations associated with those attack signatures. Accordingly, each row of table 33 contains documentation associated with each attack signature. Each description includes keywords relating to the type of attack, the network protocol used, and the success or failure of the attack, for example.

Figure 3B shows a taxonomic structure 37 defining relationships 39 of generalization between the valued attributes contained in the table 33. The taxonomic structure 37 is organized according to expert knowledge on the basis of the keywords of the documentation of the signatures in the table 33. Note that the attack signatures 31 constitute the most specific valued attributes.

The domain of the attackers valued attribute includes IP addresses. External IP addresses can be generalized by the name of the proprietor organization of the range of IP addresses to which the address belongs. The name of the organization corresponds to the field "netname" contained in the databases of the IANA™, which is the organization which oversees IP address assignment.

Internal IP addresses and private (non-routable) IP addresses can be generalized in the form of local network identifiers defined by an administrator of the intrusion detection system 1.

Finally, the names of the organizations can be generalized in the form of the value "ext" and the identifiers of the local networks can be generalized in the form of the value "int".

The domain of the victim valued attribute includes IP addresses. These IP addresses of victims can be generalized in the form of the address of the corresponding local area network.

These IP addresses can also be generalized in the form of machine names obtained by name resolution mechanisms. The machine names can be generalized in the form of host "functions" (for example the web server function) defined by the site administrator. The machine names can be generalized in the form of local area network identifiers (for example DMZ) defined by the network administrator.

The domain of the date valued attribute includes the date and time of the alerts in the format DD-MM-YYYY hh:mm:ss. The dates are successively generalized in the form minute, hour, day, month, year. These generalizations finally correspond to increasingly coarse abstractions of the time and date of an attack.